

CLAIMS

1. A multimedia scheduler for scheduling media items for network broadcast; comprising:

at least one work manager for each channel serviced, said work manager
5 including at least one producer thread, a task queue and at least one worker thread;
and

one or more scheduler objects associated with each producer thread;

wherein said work manager and associated scheduler objects create and
maintain a broadcast schedule for each channel according to predefined criteria; and

10 wherein said system is scalable to service a plurality of broadcast channels
and/or services simultaneously.

2. The scheduler of Claim 1, further comprising:

at least one property manager for providing access to all application properties;

15 at least one database connection manager for managing access to data
sources;

at least one communication module for accepting requests on a server socket
and calling appropriate routines to fulfill the requests;

at least one rules factory;

20 at least one utility thread;

at least one main application thread for initializing system components;

at least one logging system;

at least one work creator factory;

at least one data cache;

25 at least one properties file; and

at least one database.

3. The scheduler of Claim 2, wherein said predefined criteria comprise any of
programming rules, a rotation pattern for each channel, and rotation codes assigned
30 to individual media items.

4. The scheduler of Claim 3, wherein said media items comprise digital
recordings of songs.

5. The scheduler of Claim 2, wherein said rules comprise objects that implement a rules interface.

6. The scheduler of Claim 5, wherein said rules factory maintains a cache of said rule objects in memory.

7. The scheduler of Claim 2, wherein said property manager provides access to application properties, said application properties stored in any of a property file and a configuration database.

8. The scheduler of Claim 7, wherein properties are loaded into a properties object in said property manager when said property manager is initialized by said main application thread.

9. The scheduler of Claim 7, wherein said property manager includes a method for refreshing properties at regular intervals, wherein said method is called by one of said utility threads.

10. The scheduler of Claim 2, wherein said database connection manager comprises an access point to connect to said database.

11. The scheduler of Claim 9, wherein said database connection manager maintains a thread-local cache of connections for each thread.

12. The scheduler of Claim 9, wherein said database connection manager provides a connection pool to each data source.

13. The scheduler of Claim 9, wherein said database connection manager provides a thread-local cache of connections for each thread.

14. The scheduler of Claim 12, wherein said database connection manager includes methods for:

getting connections;
returning connections; and

flushing a thread local cache.

15. The scheduler of Claim 12, wherein said connection pool performs a sanity
check on each connection requested from said pool by created a dummy prepared
5 statement, and executing it.

16. The scheduler of Claim 15, wherein said connection pool discards a
connection if an exception is thrown.

10 17. The scheduler of Claim 12, wherein the connection pool opens a new
connection if one is unavailable from the pool.

18. The scheduler of Claim 2, wherein said communication module accepts
requests on a server socket and calls appropriate routines to fill the request.
15

19. The scheduler of Claim 18, wherein a port number for said socket is configured
in said property file.

20. The scheduler of Claim 18, wherein said requests comprise XML (extensible
20 markup language) strings.

21. The scheduler of Claim 18, wherein said requests include strings for:
stop;
abort;
25 channel_schedule;
status;
ping;
rotate_logfile; and
refresh.

30 22. The scheduler of Claim 2, wherein said utility threads include:
a property refresher thread;
a GC (garbage collection) thread for memory management; and
a logfile rotator thread.

23. The scheduler of Claim 22, wherein a sleep interval for each utility thread is configured in said property file.

5 24. The scheduler of Claim 2, wherein said logging system records transactions and activities.

25. The scheduler of Claim 24, wherein logging behavior is configurable by means of a logging properties file.

10

26. The scheduler of Claim 2, wherein scheduling objects comprise work creator objects and wherein associating a scheduling object with a producer thread comprises wrapping a work creator object with a producer thread

15 27. The scheduler of Claim 26, wherein said work creator objects encapsulate song and/or ad scheduling logic, wherein said work manager is independent of said song and/or ad scheduling logic.

20 28. The scheduler of Claim 27, wherein said scheduling logic includes processes for any of:

scheduling songs on stream-based channels;
scheduling ads on stream-based channels;
scheduling songs on clip-based channels; and
scheduling ads on clip-based channels.

25

29. The scheduler of Claim 28, wherein said ads are provided by a third party.

30. The scheduler of Claim 28, wherein said ads are provided in buckets, wherein a bucket is associated with a channel.

30

31. The scheduler of Claim 27, wherein said work creator objects create work objects representing tasks to be performed.

32. The scheduler of Claim 27, wherein work creator objects are created by at least one work creator factory, wherein said work creator factory is created during initialization of said work manager.

5 33. The scheduler of Claim 26, wherein said work manager further comprises at least one delete symlink (symbolic link) thread.

34. The scheduler of Claim 26, wherein said work manager further comprises at least one monitoring thread.

10

35. The scheduler of Claim 26, wherein said at least one producer thread checks a channel at configurable intervals and increments said channels schedule by generating a work request and placing it in said queue.

15 36. The scheduler of Claim 35, wherein said worker threads execute said work requests.

37. The scheduler of Claim 2, wherein said data cache caches data common to all scheduling objects and data that change infrequently, and wherein a static data cache
20 object contains a concrete data cache instance for each data source defined on a system.

38. The scheduler of Claim 37, wherein a data cache refresher thread reloads cached data, and wherein a refresh interval is specified in said property file.

25

39. The scheduler of Claim 2, wherein said main application thread comprises a starting point for said multimedia scheduler.

40. The scheduler of Claim 39, wherein said main application thread initializes
30 main components within the multimedia scheduler.

41. The scheduler of Claim 39, wherein said main application thread waits for a 'stop' or 'abort' request after initializing said main components.

42. The scheduler of Claim 39, wherein said main application thread includes a monitor, wherein a socket listener thread notifies said monitor when a request is received by a server socket.

5 43. The scheduler of Claim 42, wherein an 'abort' request stops said scheduler immediately.

44. The scheduler of Claim 42, wherein a 'stop' request allows the scheduler to stop gracefully so that:

10 work managers are stopped;
a cleanup of the database connection manager is performed; and
the application is exited.

15 45. The scheduler of Claim 39, wherein said main application thread includes a plurality of public static methods to control and report on the scheduler.

46. The scheduler of Claim 39, wherein said main application thread constitutes a class.

20 47. The scheduler of Claim 2, wherein said properties file controls behavior of said scheduler and specifies parameters to tune said scheduler.

48. The scheduler of Claim 47, wherein said properties file specifies at least:
work manager properties;
25 scheduler object properties;
scheduler-wide properties;
rules module properties;
property manager properties; and
database connection pool properties;

30 49. The scheduler of Claim 2, wherein said database comprises a plurality of tables, and wherein said at least one work manager and said property manager access said database.

50. The scheduler of Claim 49, wherein said tables include tables for at least:
current state of scheduler wherein said state is persistently saved;
an exception log for recording events, wherein events include rule violations;
enumeration of log messages;
5 associating rotation patterns to a channel;
rotation patterns created by DJ's;
rule definitions;
associating rules to a channel;
defining channel specific rule values; and
10 channel configset.

51. The scheduler of Claim 50, wherein said tables further include:
channels;
channel-specific ads;
15 channel-specific playlists;
playlists;
playlists of clips;
clips;
encoding formats;
20 song location; and
song metadata.

52. The scheduler of Claim 1, wherein said scheduler creates schedules for clip-
based channels.

25

53. The scheduler of Claim 1, wherein said scheduler creates schedules for
stream-based channels.

54. The scheduler of Claim 1, wherein said services are broadcast at different bit
30 rates.